7 Security Considerations

This specification considers two sets of security requirements, those of the applications that use the WS-Eventing protocol and those of the protocol itself.

This specification makes no assumptions about the security requirements of the applications that use WS-Eventing. However, once those requirements have been satisfied within a given operational context, the addition of WS-Eventing to this operational context can not undermine the fulfillment of those requirements; the use of WS-Eventing SHOULD NOT create additional attack vectors within an otherwise secure system.

The material below is not a "check list". There are many other security concerns that need to be considered when implementing or using this protocol. Implementers and users of this protocol are urged to perform a security analysis to determine their particular threat profile and the appropriate responses to those threats.

7.1 Notifications

The information contained in Notifications might be sensitive. In such cases it is advisable to authenticate and authorize subscribers and perform the appropriate authorization checks as part of the processing of the Subscribe request. Note that an Event Source may authorize the delivery ofsome Notifications streams might require per-Notification authorization checkson a per-message basis after the subscription has been created. This couldmight be necessary in cases where the sensitivity of the Notification information is not known at Subscribe time or might varyies over the lifetime of a subscription.

As with any message sent over a network, To protect the Notifications sent over a network, Notifications ought to have the proper authenticity, integrity and confidentiality protections applied in proportion to the sensitivity of the information they contain. In most cases such protections ought to include the authentication information necessary to allow the Event Sink to ascertain that the Notifications originated from an authorized entity (for example, the Event Source).

The ability to subscribe on behalf of a third-party Event Sink could be misused by a malicious Subscriber to create a denial-of-service attack. While it does not remove the ability for such misuse, authenticating Subscribers provides a way to deter and trace the origin of such attempts. Additionally, the authorization of Subscribers reduces the pool of potential attackers.

7.2 Subscriptions

Once created, subscriptions to sensitive Notification streams ought to be treated as protected resources. Renew, GetStatus, and Unsbubscribe requests ought to be authenticated and authorizedthe identity of the requester ought to be checked against the “sink owner” of the subscription (for example, the identity of the requester ought to be checked against the identity of the entity that performed the original Subscribe request). Likewise SubscriptionEnd messages ought to be authenticated and authorizedverified to originate from the “source owner” of the subscription (for example, the identity of the sender ought to be checked against the identity of the entity that sent the original SubscribeResponse message). Note that determinations of subscription ownership areauthentication and authorization policies (i.e.i.e. the rules that define which entities are allowed to perform which requests and the

mechanisms by which the identities of these entities are discovered and verified) are particular to individual deployments. ~~For example, within a particular organization it might be the rule that "source owner" of a subscription is considered to be the Event Source (i.e. the entity that processes the original Subscribe request) of that subscription whereas, in another organization, the "source owner" could be the Subscription Manager.~~

7.3 Endpoint Verification

Implementations that perform validity checks on the EPRs used in WS-Eventing (wse:NotifyTo, wse:EndTo) are advised that such checks can be misused to obtain information about a target network. For example, suppose an Event Source~~ink~~ implementation verifies the address of NotifyTo EPRs by attempting to create a connection to ~~this~~the EPR's address and faulting the Subscribe request if ~~such~~ a connection cannot be created. When deployed within a DMZ, such an Event Source~~ink~~ could be exploited by a malicious Subscriber to probe for other, non-visible ~~machines~~hosts by guessing target addresses ~~values~~ and using ~~these values~~them in Subscribe requests. Note that, even if the returned fault does not provide ~~enough information to enable such attacks~~connection information, the time the Event Source~~ink~~ spends processing the Subscribe request might ~~betray~~reveal ~~if the~~there is a host ~~existence or non-existence of a host~~ with ~~at~~ the target address.

Implementations that perform validity checks on the EPRs used in WS-Eventing are advised to provide a means to disable such checks in environments where these types of attacks are an issue.